Replace paragraph no. 7 with the following:

[0007] To further burden the security engineer, each vulnerability or potential intrusion needs to be identified and a description of it stored for use by the vulnerability or intrusion detection tools. This process, however, is often complicated. For instance, it is extremely difficult just to write an application that would check a Secure Shell (SSH) server to determine if the remote system was running a version of SSH that is vulnerable to a Denial of Service attack. Traditional development methodologies force the user to have an intimate understanding of TCP/IP and a low-level (often cumbersome) development language such as ANSI C or Perl. Even advanced "Attack Scripting Languages" are overly cumbersome and require an understanding of variables, "for" loops, "while" loops, and other development syntax.

Replace paragraph no. 11 with the following:

[0011] An IDS used with an embodiment of the invention monitors network traffic for signs of malicious activity. This analysis of network traffic is also based on a set of rules. However, the rules used by the IDS are determined based on the analysis of the network performed by the VDS – the IDS only monitors for intrusive traffic that can actually affect the particular network.

Replace paragraph no. 26 with the following:

[0026] The rules referred to above that are stored in the rules database or loaded into the IDS are query-based, and resemble "assertions" or "queries" found in typical SQL. The rules are structured to be assertions that, if found true, identify the presence of a particular condition, such as an operating system, application, vulnerability, or intrusion. Hence, collectively, these rules serve to identify and name the characteristics and properties of the network 100. For instance, to test for a vulnerability in the Line Printer Daemon (LPD) that shipped with the